doi:10.11959/j.issn.1000-436x.2016119

SDN 多控制器一致性的量化研究

李军飞，兰巨龙，胡宇翔，邬江兴
（国家数字交换系统工程技术研究中心，河南 郑州 450002）

摘 要：针对 SDN 网络中多控制器的一致性问题，提出了一种量化的研究方法，为控制层的东西向扩展提供更为精准有效的共享网络视图方法。首先，结合 SDN 的特性，给出了控制器之间一致性、性能以及可用性的度量指标，建立通用的量化分析模型。其次，针对其中 3 类典型的一致性问题进行了量化研究，明确了其取得最优值的条件，为一致性参数的配置提供了参考。最后，通过仿真实验对该量化方法进行验证。实验结果表明，该量化方法能够有效提高 SDN 控制层的性能和可用性。
关键词：SDN；多控制器；一致性；性能；可用性
中图分类号：TP393 **文献标识码**：A

Quantitative approach of multi-controller’s consensus in SDN

LI Jun-fei, LAN Ju-long, HU Yu-xiang, WU Jiang-xing
(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: For the problem of multi-controller’s consensus in SDN, a quantitative approach was proposed, which provided a more accurate and effective method of sharing network view for the control layer's east-west extension. Firstly, the metrics of consensus, performance and availability between the controllers with the feature of SDN was provided, establishing the common model for quantitative research. Secondly, for the three typical questions in the research of multi-controller’s consensus, the condition to achieve its optimal value was explicated, which provided a reference for the configuration of consensus. Finally, to verify the validity of the quantitative approach by simulation, experimental results show that this approach can improve the performance and availability of the control layer in SDN effectively.
Key words: SDN, multi-controller, consensus, performance, availability

1 引言

随着网络应用的快速发展，传统的网络交换设备承载着越来越多的控制逻辑，已难以适应虚拟化、云计算、大数据及相关业务发展对数据高速传输、资源灵活配置、协议快速部署的需求。软件定义网络（SDN, software defined network）提出了控制与转发分离的设计结构，实现了开放的可编程网络接口，为网络提供了更细粒度的管理，引起了学术界和产业界的广泛研究[1,2]。然而，SDN 的集中式控制在为网络应用带来创新与便利的同时，也带来了可靠性、可扩展性以及可用性等方面的问题。目前，无论是针对 SDN 的可靠性问题[3]，还是可扩展性问题[4]，大多数的解决方法均采用多控制器间主备冗余或对等协同的方法。

然而，多个控制器之间如何高效地共享网络视图，以期实现快速的主备切换或有效的集中式控制，即 SDN 控制器的一致性(consensus)问题，仍是 SDN 多控制器网络面临的主要难题之一。维护 SDN 控制器一致性的主要目的就在于要保证主控制器

收稿日期：2015-10-19；**修回日期**：2016-05-10
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（No.2012CB315901, No.2013CB329104）；国家自然科学基金资助项目（No.61521003，No.61372121）；国家高技术研究发展计划（“863”计划）基金资助项目（No.2015AA016102, No.2013AA013505）
Foundation Items: The National Basic Research Program of China(973 Program) (No.2012CB315901, No.2013CB329104), The National Natural Science Foundation of China (No.61521003, No.61372121), The National High Technology Research and Development Program of China(863 Program) (No.2015AA016102, No.2013AA013505)

获取的网络事件能够共享给从控制器，或本地控制器获取的网络事件能够传播给全局其余的控制器，以使多个控制器在关于全网的视图问题上达成一致。在 SDN 网络中，强一致性有利于多个控制器间具有更加一致的网络视图，使基于集中式控制的上层应用的效果更好。但是，这也会带来更多的通信开销和延迟，影响控制层面的性能并降低其可用性。

目前，SDN 控制器的一致性主要是通过分布式数据存储系统来实现[5]，其可供选择的一致性协议、数据交互和同步方法有限，难以满足上层应用的多样化部署和流量优化需求[6,7]。而且，控制器一致性的研究主要集中在设计和实现方面，多是针对不同强度一致性的定性研究，缺少对其进一步的量化和优化。因此，研究多控制器之间一致性、性能以及可用性的量化关系，并基于此来寻求 SDN 多控制器间的协同优化配置，对于提升 SDN 控制层面的整体性能具有重要的现实应用价值。

综上，本文提出了一种 SDN 多控制器的一致性量化研究方法，希望能引起相关研究者的兴趣。具体而言，本文通过建立 SDN 多控制器的通用量化分析模型，研究其一致性、性能及可用性之间的协同优化配置，以减少同步的通信开销、提高可用性。

2　相关工作

SDN 的主要优势之一就是具有全局的网络视图，供部署在控制器上层的应用所使用，以简洁高效地解决传统网络中难以解决的问题，如 Handigol、Wang 等[1,2]的研究结果表明集中式网络具有更为有效的负载均衡管理。然而，SDN 网络中也存在着可靠性、可扩展性以及可用性等方面的问题，成为了该领域的研究热点。例如，Li 等[3]针对主控制器异常会造成整个网络瘫痪的问题，提出了多个冗余控制器采用 BFT (byzantine fault tolerant)技术，以增强控制层的可靠性。另外，Dixit 等[4]针对不断增长的 SDN 网络规模及单个控制器性能有限的问题，提出了多个分布式控制器协同工作的方法，以解决 SDN 网络的可扩展性。因此，无论是解决 SDN 网络中的可靠性还是可扩展性，目前大多数的方法都是基于多控制器的思想。然而，由于不同控制器管理的交换机不同或角色不同，导致其获得的网络视图不同，为了能够达到 SDN 的集中式控制效果，多个控制器之间需要同步其网络视图。控制器之间如何同步，使具有更加一致的网络视图，是上述研究要解决的根本问题。

HyperFlow[8]是第一个提出在 SDN 网络中引入分布式控制器概念的，控制器之间通过构建在分布式文件系统 WheelFS 之上的订阅—发布平台实现数据同步，以确保节点之间对网络视图的一致性。同时，文献[8]也评估了 HyperFlow 中控制器节点之间的同步性能，即每秒能够处理 1 000 次以下的网络事件更新，但没有对一致性强弱对性能的影响做进一步的研究。Onix[5]是第一款产品级的强调可用性和可扩展性的 SDN 控制器，由于其针对大规模的商用级网络所开发，所以广泛应用于 Google、VMware 等公司的商用网络中。Onix 控制器为上层应用提供了 NIB (network information base) 的网络视图，并有 2 种可供选择的不同类型的数据库用于节点间的数据同步：面向 SQL 的、具有强一致性的事务型数据库，但其同步性能较低；基于 DHT (distributed Hash map) 的、仅能最终一致性的 Key-Value 数据库，但有较好的同步性能。然而，Onix 是一款不开源的控制器，且仅提供了 2 种不同强弱的一致性模型，难以适应 SDN 上层应用的多样化需求。OpenDayLight 是一个以商用为初衷的开源的控制器项目，得到 Cisco、IBM 等公司的支持，其中的多控制器协同主要是采用云计算中的 Infinispan 数据存储框架[9]，但由于其目前仍在开发之中，没有达到理想的性能指标，故需要重新考虑该部分的设计[10]。

另外，Bailis 等[11]研究了如何采用 PBS (probabilistically bounded staleness) 模型预测分布式节点间数据的最终一致性强度，为一致性和性能之间的权衡提供了借鉴方法。但是，该模型针对通用的分布式系统，没有结合 SDN 的特性给出具体的优化方法。Hassas 等[12]提出了一种层次式架构来实现多控制器间的一致性，其中，上下两层控制器形成树状结构，下层的控制器负责局部交换机的管理，上层控制器协调下层控制器的同步，维护网络视图的一致性。然而，该方法仅改变了多控制器间的通信模式，并没有缓解一致性和性能之间的冲突问题。

3　一致性问题的量化模型

3.1　动机

在分布式系统一致性的研究中，一致性与性能之间的平衡始终是研究热点，针对不同的应用场景

提出了不同强弱的一致性协议[13,14]。例如，Paxos 是解决分布式系统一致性问题的经典算法，可以保证多控制器之间具有强一致的网络视图，但是在一次数据同步中，proposer、acceptor 和 learner 之间需要多次交互。如图 1 所示，这种信息交互方式会产生较大的网络通信开销，尤其是在选举失败或提议冲突的情况下消耗的网络带宽更为严重。因此，采用弱一致性协议或最终一致性协议来降低节点间同步的通信开销，对于提升性能具有较好的效果。然而，弱一致性导致了控制器之间网络视图的差异，影响某些上层应用的功能和效果，文献[6]中所提及的流量负载均衡应用。因此，如何在满足上层应用一致性要求的情况下，实现最小的同步通信开销以及最高的可用性，是本文的研究重点。

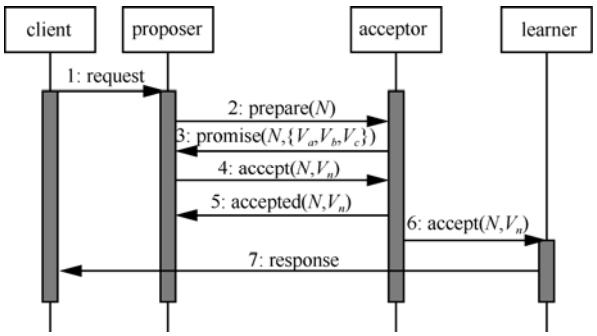

图 1 Paxos 算法的处理流程

3.2 相关概念

3.2.1 一致性

对于一个 SDN 控制器节点，由于本地缓存或通信延迟，使其控制区域内更新的网络事件没有及时地共享给它其余的控制器，造成了节点间网络视图的不一致。因此，这里采用控制器间网络事件的差异度作为一致性的度量指标，控制器之间网络事件的差异度越大，意味着系统的一致性越弱。另外，控制器节点上有不同类型的网络事件，表 1 中列举了部分常见的事件，如新的主机节点加入、链路断开、流量负载更新等，并且，一致性的强弱很大程度上依赖于具体的应用，例如，对于路径计算应用来说是弱一致性的协议，而对于负载均衡应用来说却可能是强一致性的。同时，不同的网络事件对于同一上层应用的一致性影响是不同的，例如，对于路径计算应用的一致性而言，网络拓扑的变化相对于流量负载的变化，具有更大的影响。

表 1 常见的网络事件

Host	Port	Node	Link	Traffic
Active	Shut	Normal	Connect	Increase
Inactive	Open	Unnormal	Disconnect	Decrease

综上可知，一致性的分析是要针对某一具体的上层应用，且取决于该应用中所涉及网络事件的类型和数量。所以，对于部署在多个 SDN 控制器上的某一应用 A，定义控制器节点 CN_i 的一致性为

$$c_i = \sum_{j=1}^{m} \lambda_j \left| \Delta E_{ij} \right| \tag{1}$$

其中，E_{ij} ($1 \leqslant j \leqslant m$)表示在控制器节点 CN_i 上应用 A 中所涉及的网络事件的类型，$|\Delta E_{ij}|$表示该控制器节点上已更新的但还没有及时同步的网络事件 E_{ij} 的最大数量，λ_j 表示事件 E_{ij} 在应用 A 中的影响因子。可以看出，c_i 的取值越大，该节点的一致性越弱。进一步地，可以定义应用 A 在整个系统中的一致性为

$$C = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_j \left| \Delta E_{ij} \right| \tag{2}$$

其中，n 表示网络中控制器的数目。那么，当仅有 c_i=1，$\forall j \neq i$，$c_j = 0$，意味着 SDN 控制层面仅有一个控制器节点有事件更新，对应用于增强 SDN 网络可靠性的主备冗余多控制器网络；当 $\forall i$，c_i=1，意味着每个节点最多有一个需要同步的事件，对应强一致性的分布式多控制器网络。

3.2.2 性能

控制器之间进行一致性同步的性能主要受限于 2 个因素：1）节点间的通信开销；2）单节点的同步负载。下面将逐一分析上述因素与一致性的关系，以研究性能与一致性之间的平衡点。

对于通信开销，控制器节点 CN_i 在同步一个网络事件 E_{ij} 时，产生的数据分组通常较小，平均在 1 KB 左右，最大不超过 4 KB[15]，如图 1 中步骤 2 的数据分组。而且，一致性协议交互过程中的其余数据分组的大小也大都在该范围内，差异不大，如图 1 中步骤 3~步骤 5 通信过程的数据分组。因此，使用单位时间内的通信次数作为衡量通信开销的指标，文献[16]中也采用了该近似方法。那么，当控制器节点 CN_i 每次均在达到一致性上限 C_i 时，节点间进行一次一致性同步，则可以获得最小的通信开销，速率为

$$v_i = \frac{\rho(n-1)\sum_{j=1}^{m}\lambda_j g_{ij}}{c_i} \tag{3}$$

其中，ρ是一个固定的常数，取决于系统中采用的一致性协议。g_{ij}表示控制器节点 CN_i上网络事件 E_{ij}的产生速率。进一步，可以计算整个系统通信开销的最小速率为

$$V = \rho(n-1)\sum_{i=1}^{n}\frac{\sum_{j=1}^{m}\lambda_j g_{ij}}{c_i} \tag{4}$$

单节点的同步负载主要是指本地发送的或接受远程的同步请求，可以采用同步次数作为其量化指标，即上文中所述的 V_i。因此，式(3)和式(4)可以表示一致性与性能之间的关系。

3.2.3　可用性

可用性是另一个与一致性冲突的因素，在要求强一致性的多控制器网络中，当某个节点出现故障时，使控制器之间的网络视图形成差异。此时，控制层由于无法达成强一致性的网络视图，将不能再继续工作，失去可用性。控制器节点间如果减弱一致性强度，在本地缓存部分更新的网络事件，则能够容忍短时间的故障。

这里采用与文献[16,17]中类似的方法，定义可用性为$\frac{n_{\text{cap}}}{n_{\text{sub}}}$，其中，$n_{\text{cap}}$表示可以接受更新事件的数量，$n_{\text{sub}}$表示已经提交的更新事件的数量。$n_{\text{sub}}$取决于上层应用 A 中所涉及的网络事件的更新速率，与一致性参数的配置无关。因此，本文的研究集中在 n_{cap}，同样需要考虑不同类型事件的影响力。对于控制器节点 CN_i，在满足一致性约束的条件下，n_{cap}的最大取值为 C_i。另一方面，考虑到故障节点的修复时间 t_f，则 n_{cap}的最大取值为 $t_f\sum_{j=1}^{m}\lambda_j g_{ij}$，所以

$$n_{\text{cap}} = \text{Min}\{c_i, t_f\sum_{j=1}^{m}\lambda_j g_{ij}\} \tag{5}$$

进而，整个控制层的可用性可以被度量为

$$Ava = \sum_{i=1}^{n}\text{Min}\{c_i, t_f\sum_{j=1}^{m}\lambda_j g_{ij}\} \tag{6}$$

3.3　一致性问题

在对 SDN 多控制器的一致性、性能以及可用性进行量化分析之后，接下来本文研究它们之间的平衡点，以获取最大的效益。在对某一上层应用进行一致性配置时，通常会有 2 类针对不同目标的优化问题：1）在给定一致性的约束条件下，求解可以实现的最大性能或最高可用性；2）在给定性能或可用性约束的条件下，求解可以实现的最大一致性。对于这些约束条件，一般从全局的角度对整个系统进行约束，进一步地，考虑到网络环境和应用需求的多样性，也可以对每个控制器节点进行具体的约束。因此，该一致性问题又可以细分为几类具体的子问题，下面选取其中具有代表性的 3 个问题进行详细的描述和分析。

1) 给定一致性的全局约束，求能实现的最大性能

Q1：对某一上层应用 A，已知其在各个控制器节点上所涉及网络事件的更新速率，即$\sum_{j=1}^{m}\lambda_j g_{ij}$，在式(2)的 $C \leqslant C_b$ 一致性约束的条件下，求式(4)的最小取值 $\text{Min}\,V$。

2) 给定一致性的全局约束，求可获得的最高可用性

Q2：对某一上层应用 A，已知其在各个控制器节点上所涉及网络事件的更新速率，即$\sum_{j=1}^{m}\lambda_j g_{ij}$，以及故障节点修复时间 t_f，在式(2)的 $C \leqslant C_b$ 一致性约束的条件下，求式(6)的最大取值 $\text{Max}\,(Ava)$。

3) 给定一致性的具体约束，求能实现的最大性能

Q3：对某一上层应用 A，已知其在各个控制器节点上所涉及网络事件的更新速率，即$\sum_{j=1}^{m}\lambda_j g_{ij}$。给定式(1)的一致性约束 $\forall i, c_i \leqslant b_i$，及式(2)的约束 $C \leqslant C_b$，且$\sum_{i=1}^{n} b_i \geqslant C_b$，求式(4)的最小取值 $\text{Min}\,V$。

4　一致性问题的求解

本节将逐一讨论上述一致性问题的最优解，并分析其获得最优解的条件。

4.1　问题 Q1 的最优解

令 $x_i = \sqrt{\frac{\sum_{j=1}^{m}\lambda_j g_{ij}}{c_i}}$，$y_i = \sqrt{c_i}$，则由 Cauchy-Schwarz 不等式可得

$$(x_1^2 + \text{L} + x_n^2)(y_1^2 + \text{L} + y_n^2) \geqslant (x_1 y_1 + \text{L} + x_n y_n)^2 \tag{7}$$

$$\frac{V}{\lambda(n-1)}C \geqslant \left(\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}\lambda_j g_{ij}}\right)^2 \tag{8}$$

当式(7)中满足 $\forall i,j,i\neq j,\frac{x_i}{y_i}=\frac{x_j}{y_j}$，且式(8)中满足 $C=C_b$ 时，式(4)取得最小值

$$\mathrm{Min}\,V=\frac{\lambda(n-1)}{C_b}\left(\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}\lambda_j g_{ij}}\right)^2$$

此时，$c_i=C_b\dfrac{\sqrt{\sum_{j=1}^{m}\lambda_j g_{ij}}}{\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}\lambda_j g_{ij}}}$。

c_i 的取值虽然复杂，但其含义是简单清晰的，可以称其为“平方根分布法则”[16]，表示每个节点的一致性上限与其产生网络事件速率的平方根成正比。所以，在给定一致性的全局约束下，可以根据各个节点产生网络事件的速率来分配各节点的一致性上限，以获得最小的通信开销，实现更高的性能。

4.2 问题 Q2 的最优解

在式(6)中，可用性 *Ava* 有明确的上确界

$$\sum_{i=1}^{n}\mathrm{Min}\{c_i,t_f\sum_{j=1}^{m}\lambda_j g_{ij}\}\leqslant \mathrm{Min}\left\{\sum_{i=1}^{n}c_i,t_f\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j g_{ij}\right\}\leqslant \mathrm{Min}\left\{C_b,t_f\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j g_{ij}\right\}$$

此时，若 $c_i=C_b\dfrac{\sum_{j=1}^{m}\lambda_j g_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j g_{ij}}$，上述不等式满足“=”关系，式(6)可取得最大值

$$\mathrm{Max}(Ava)=\mathrm{Min}\left\{C_b,t_f\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j g_{ij}\right\}$$

同问题 Q1 类似，c_i 的取值取决于节点产生网络事件的速率，当两者之间构成正比关系时，整个控制层的可用性达到最大。同时，本文也注意到该问题的最优解并不唯一，上述仅是其中的一组。

4.3 问题 Q3 的最优解

问题 Q3 的求解是一个典型的多约束非线性规划问题，可形式化描述为

$$\begin{cases}\mathrm{Min}\ V(\vec{\boldsymbol{x}})\\ \text{s.t.}\ A\vec{\boldsymbol{x}}\leqslant \vec{\boldsymbol{b}}\\ g(\vec{\boldsymbol{x}})\leqslant d\\ g(\vec{\boldsymbol{x}})\geqslant d\end{cases} \tag{9}$$

其中，$\vec{\boldsymbol{x}}=[c_1,\mathrm{L}\ ,c_n]^{\mathrm{T}}$，$\vec{\boldsymbol{b}}=[b_1,\mathrm{L}\ ,b_n]^{\mathrm{T}}$，$d=C_b$，$A=E_{n\times n}$，$g$ 表示向量中项的求和。式(9)的常规解法不仅复杂，计算量较大，而且不能保证是全局最优解。因此，针对 $V(\vec{\boldsymbol{x}})$ 的特殊性，设计了一个近似最优解的启发式算法，以期大幅度降低该问题的计算复杂度。

MNP 启发式算法

输入 整体约束 C_b，具体约束 $\vec{\boldsymbol{b}}$，各节点的流量 $t_i=\sum_{j=1}^{m}\lambda_j g_{ij}$

输出 一致性的上限 $\vec{\boldsymbol{x}}$

1) $T=C_b$，$X=\{x_1,\cdots,x_n\}$;
2) **while** $X\neq\phi$
3) $\quad I\leftarrow\phi$;
4) $\quad$ **foreach** $x_i\in X$
5) $\quad\quad x_i'=T\dfrac{\sqrt{t_i}}{\sum_{x_j\in X}\sqrt{t_j}}$;
6) $\quad\quad$ **if** $x_i'>b_i$
7) $\quad\quad\quad I\leftarrow I\cup x_i$;
8) $\quad$ **if** $I=\phi$
9) $\quad\quad$ **foreach** $x_h\in X$
10) $\quad\quad\quad x_h=x_h'$;
11) $\quad\quad X\leftarrow\phi$;
12) $\quad$ **else**
13) $\quad\quad x_m$ s.t. $x_m\in I$ and $\forall x_i\in I\ t_m\geqslant t_i$;
14) $\quad\quad x_m=b_m$;
15) $\quad\quad T\leftarrow T-b_m$;
16) $\quad\quad X\leftarrow X-x_m$

在该启发式算法中，首先按照 4.1 节中的平方根分布法则，计算在全局约束下取得最优解的点 $\vec{\boldsymbol{x}}'$（第 4)、5)行），以其作为启发，搜索临近的近似最优解。同时，记录与局部一致性相冲突的项 x_i，构成集合 I（第 6)、7)行）。然后，在能够较快接近最优解的方向上搜索，即在集合 I 中选择 x_m，使其对应的 t_m 是所有冲突项中最大的，并使其满足局部一致性约束（第 13)、14)行）。最后，从解向量中去除 x_m 项（第 15)、16)行），进行下一轮的最优解搜索，直至所有的项同时也满足局部约束（第 8)~11)行）。

5 实验验证与分析

5.1 测试环境

为了测试该量化模型，本文采用 C++语言实现

了一个仿真器，来模拟多控制器之间的一致性交互。该仿真器基于 Internet 2 OS3E 网络拓扑，OS3E 是一个遍布美国的用于先进科学研究的 SDN 网络[18]。如图 2 所示，OS3E 具有 34 个节点和 42 条链路，每个节点表示一个独立的大学或组织，通常需要部署一个 SDN 控制器。因此，在该仿真器中，本文假定 OS3E 的每个节点上均部署一个控制器，控制器之间需要一致性交互。

图 2　OS3E 的网络拓扑

对于 OS3E 中各节点网络事件的更新速率 $\sum_{j=1}^{m}\lambda_j g_{ij}$，本文以斯坦福大学校园网中的统计数据作为参考[19]，如图 3 所示，其中，不再对具体的事件类型加以区分。图 3 反映了一天中控制器负载的大致分布，仿真器中以其作为各节点网络事件的更新速率随时间的变化趋势，且各节点的时间以其所处的时区为准（西五区至西八区）。另外，仿真器中考虑了 2 种不同的场景：1) OS3E 网路是同构的，即各节点网络事件的平均更新速率均为 4 900 flow/s；2) OS3E 网络是异构的，即各节点网络事件的平均更新速率在[3 000, 6 800]内随机取值。

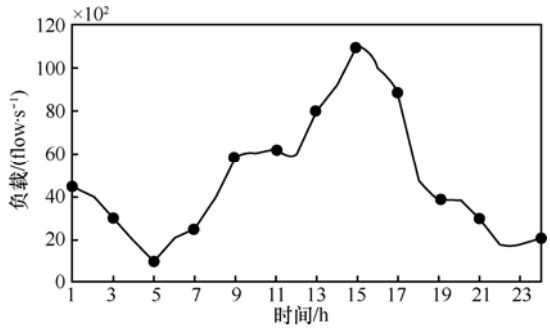

图 3　控制器的负载分布

同时，在该仿真器中实现了 3 种一致性策略：1) Strict Mode：确保 SDN 网络中的每个事件能够被及时地全局共享，即 $\forall i, c_i \leqslant 1$，类似于 Onix 中基于 SQL 的强一致性模型；2) Relaxed Mode：在满足一致性约束的前提下，节点间选择最弱的一致性协议，且各节点的一致性配置相同，即 $\forall i, c_i = \mathrm{Min}\left\{\mathrm{Min}\{b_j\}, \frac{C_b}{n}\right\}$，类似于 Onix 中基于 Key-Value 的弱一致性模型；3) Elastic Mode：基于上述量化模型的分析结果，设计一种具有弹性的一致性协议，使各节点的一致性配置与其网络事件的更新速率相协调。通过分析上述 3 种策略的仿真结果，来说明该量化模型相对于 SDN 网络中常用一致性模型的优势。

5.2　实验数据

对于要在多个控制器节点上部署的上层应用 A，假定其全局一致性约束 $C_b = 68$，并在上述实验环境中对其进行仿真测试。图 4 所示为在该约束下全局通信开销的仿真结果，包括同构网络和异构网络 2 种仿真场景，其中的 X 轴采用西八区（PST）时间。从仿真结果可以看出，在 3 种一致性策略中，Elastic Mode 一直具有最小的通信开销，Strict Mode 的通信开销最大，且三者随时间的走势大致相同。所以，Elastic Mode 相对于常用的一致性策略能够实现更好的性能。另一方面，通过对比分析具体的数据，可以发现在异构网络中，Elastic Mode 具有更好的效果，即各节点网络事件数量的差异越大，Elastic Mode 提升性能的效果越显著。

进一步地，在全局一致性约束的基础上，再对每个控制器节点做具体的约束，即 $\forall i, c_i \leqslant b_i$，其中，$b_i$ 在[1, 3]内随机取值。Elastic Mode 采用 4.3 节所述的启发式算法，计算每个节点的一致性参数 c_i。这里仅在异构网络中进行了仿真，图 5 给出了该一致性约束下的通信开销。同样，Elastic Mode 具有最小的通信开销，而 Strict Mode 的通信开销最大。另外，与图 4(b)对比，图 5 中 Relaxed Mode 通信开销的抖动较大，即在一些特殊情况下的性能较差（如图 5 中 7 h、11 h、19 h 等时刻），而 Elastic Mode 通信开销的变化较为平稳，能适应不同情况的一致性约束。

对于可用性的仿真测试，为了便于对比分析，考虑另一个在多个控制器节点上部署的上层应用 B，假定其全局一致性约束 $C_b = 1\,700$。由于各节点产生网络事件的速率很快，为了不致使其在故障发生后均能超出本地一致性约束的上限，假定系统可

以在很短的时间内修复故障，即 t_f = 0.01 s。这里也仅在异构网络中进行了仿真，测试结果如图 6 所示，Y 轴表示式(6)中的 Ava。其中，Elastic Mode 和 Relaxed Mode 的可用性大致相当，在 9~15 h 时，Elastic Mode 的可用性略占优势。而 Strict Mode 的可用性为定值，远低于上述两者的可用性。

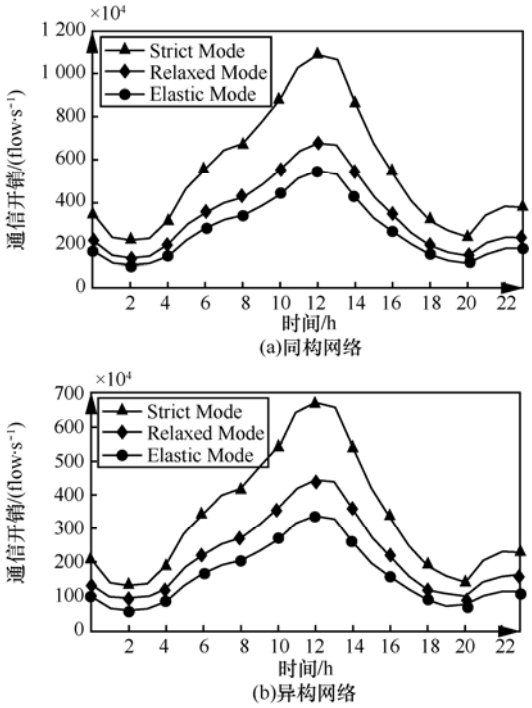

图 4　全局约束下的通信开销

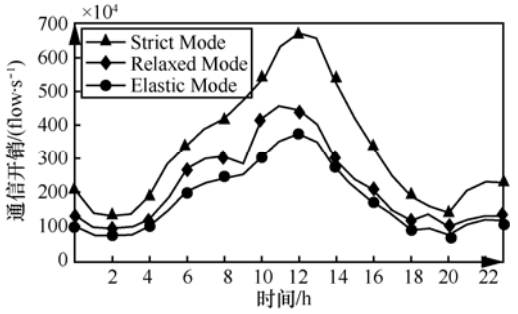

图 5　具体约束下的通信开销

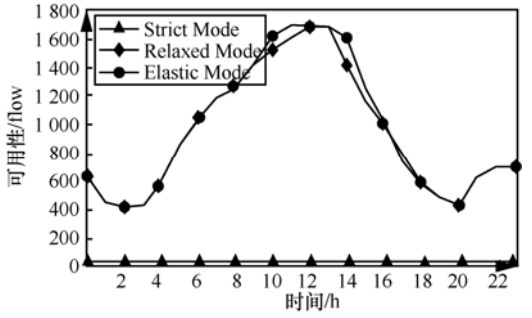

图 6　全局约束下的可用性

综合上述分析可以发现，在多控制器的 SDN 网络中，基于该量化模型来适时调整一致性参数的 Elastic Mode，相对于传统的一致性策略 Strict Mode 和 Relaxed Mode，在减小通信开销和提高可用性上，都具有一定的优势。因此，该量化模型对于研究多控制器之间的一致性问题，提升控制层面的性能和可用性，具有重要的意义。

6　结束语

本文针对 SDN 多控制器间的一致性问题，首先介绍了该领域的相关工作和研究。然后，结合 SDN 的特性，给出了一致性、性能及可用性的度量指标，建立了量化分析模型。然后，选择其中的几类一致性问题进行量化研究，求解其最优值以及获得最优值的条件，为一致性参数的配置提供了指导。最后，将基于上述量化模型的一致性配置方法与传统的一致性方法进行了对比，实验结果表明该方法能够有效提高控制层面的性能和可用性。下一步的工作是把该方法应用到 OpenDayLight 控制器上，将其作为东西向接口以实现控制器之间的同步，测试在真实 SDN 环境下的一致性效果。

参考文献：

[1] HANDIGOL N, SEETHARAMAN S, FLAJSLIK M, et al. Plugin-serve: load-balancing Web traffic using OpenFlow[J]. ACM SIGCOMM Demo, 2009, 4(5): 6.

[2] WANG R, BUTNARIU D, REXFORD J. OpenFlow-based server load balancing gone wild[C]//USENIX HotICE. c2011:12.

[3] LI H, LI P, GUO S, et al. Byzantine-resilient secure software-defined networks with multiple controllers in cloud[J]. IEEE Transactions on Cloud Computing, 2014, 2(4): 436-447.

[4] DIXIT A, HAO F, MUKHERJEE S, et al. Towards an elastic distributed SDN controller[C]//ACM SIGCOMM Computer Communication Review, 2013, 43(4): 7-12.

[5] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed control platform for large-scale production networks[C]//OSDI. c2010: 1-6.

[6] LEVIN D, WUNDSAM A, HELLER B, et al. Logically centralized: state distribution trade-offs in software defined networks[C]//The First Workshop on Hot Topics in Software Defined Networks. ACM, c2012: 1-6.

[7] STRAUß J. Control-plane consensus in software-defined networking: distributed controller synchronization using the ISIS² toolkit[J/OL]. http://elib.uni-stuttgart.de/handle/1162/357/.

[8] TOOTOONCHIAN A, GANJALI Y. HyperFlow: a distributed control plane for OpenFlow[C]//The 2010 Internet Network Management

Conference on Research on Enterprise Networking. USENIX Association, c2010: 3.

[9] LUO M, WU X, ZENG Y, et al. Multi-dimensional hashing for fast network information processing in SDN[C]//Complex, Intelligent, and Software Intensive Systems (CISIS), 2015 Ninth International Conference. IEEE, c2015: 140-147.

[10] BOTELHO F, BESSANI A, RAMOS F, et al. SmartLight: a practical fault-tolerant SDN controller[J]. arXiv preprint arXiv:1407.6062.

[11] BAILIS P, VENKATARAMAN S, FRANKLIN M J, et al. Probabilistically bounded staleness for practical partial quorums[J]. Proceedings of the VLDB Endowment, 2012, 5(8): 776-787.

[12] HASSAS Y S, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]//The First Workshop on Hot Topics in Software Defined Networks. ACM, c2012: 19-24.

[13] BAILIS P, VENKATARAMAN S, FRANKLIN M J, et al. Quantifying eventual consensus with PBS[J]. The VLDB Journal, 2014, 23(2): 279-302.

[14] DIAO Z. Consistency models for cloud-based online games: the storage system's perspective[J/OL]. http://ceur-ws.org/Vol-1020/paper_03.pdf.

[15] BOTELHO F, RAMOS V, MANUEL F, et al. On the feasibility of a consistent and fault-tolerant data store for SDNs[C]//Software Defined Networks (EWSDN), 2013 Second European Workshop. IEEE, c2013: 38-43.

[16] ZHANG C, ZHANG Z. Trading replication consensus for performance and availability: an adaptive approach[C]//Distributed Computing Systems, 23rd International Conference. IEEE, c2003: 687-695.

[17] YU H, VAHDAT A. The costs and limits of availability for replicated services[J]. ACM SIGOPS Operating Systems Review, ACM, 2001, 35(5): 29-42.

[18] Internet2 open science, scholarship and services exchange[EB/OL]. http://www.internet2.edu/network/ose/.

[19] SHALIMOV A, ZUIKOV D, ZIMARINA D, et al. Advanced study of SDN/OpenFlow controllers[C]//The 9th Central & Eastern European Software Engineering Conference in Russia. ACM, c2013: 1.

作者简介：

李军飞（1989-），男，河南安阳人，国家数字交换系统工程技术研究中心博士生，主要研究方向为集中式网络管控下的主动防护技术。

兰巨龙（1962-），男，河北张北人，博士，国家数字交换系统工程技术研究中心总工程师、教授、博士生导师，主要研究方向为新一代信息网络关键理论与技术。

胡宇翔（1982-），男，河南周口人，博士，国家数字交换系统工程技术研究中心讲师，主要研究方向为新一代信息网络关键理论与技术。

邬江兴（1953-），男，浙江嘉兴人，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为网络通信和网络安全。